```
RRRRRRRRRRR      UUU          UUU  NNN          NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN          NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN          NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRR          RRR UUU          UUU  NNN          NNN  000         000  FFF              FFF
RRR          RRR UUU          UUU  NNN          NNN  000         000  FFF              FFF
RRR          RRR UUU          UUU  NNN          NNN  000         000  FFF              FFF
RRR          RRR UUU          UUU  NNNNNN       NNN  000         000  FFF              FFF
RRR          RRR UUU          UUU  NNNNNN       NNN  000         000  FFF              FFF
RRR          RRR UUU          UUU  NNNNNN       NNN  000         000  FFF              FFF
RRRRRRRRRRRR     UUU          UUU  NNN  NNN     NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN  NNN     NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN  NNN     NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRR    RRR       UUU          UUU  NNN       NNNNNN  000         000  FFF              FFF
RRR    RRR       UUU          UUU  NNN       NNNNNN  000         000  FFF              FFF
RRR    RRR       UUU          UUU  NNN       NNNNNN  000         000  FFF              FFF
RRR       RRR    UUU          UUU  NNN          NNN  000         000  FFF              FFF
RRR       RRR    UUU          UUU  NNN          NNN  000         000  FFF              FFF
RRR       RRR    UUU          UUU  NNN          NNN  000         000  FFF              FFF
RRR          RRR UUUUUUUUUUUUUUU  NNN          NNN     000000000     FFF              FFF
RRR          RRR UUUUUUUUUUUUUUU  NNN          NNN     000000000     FFF              FFF
RRR          RRR UUUUUUUUUUUUUUU  NNN          NNN     000000000     FFF              FFF
RRR          RRR UUUUUUUUUUUUUUU  NNN          NNN     000000000     FFF              FFF
RRR          RRR UUUUUUUUUUUUUUU  NNN          NNN     000000000     FFF              FFF
```

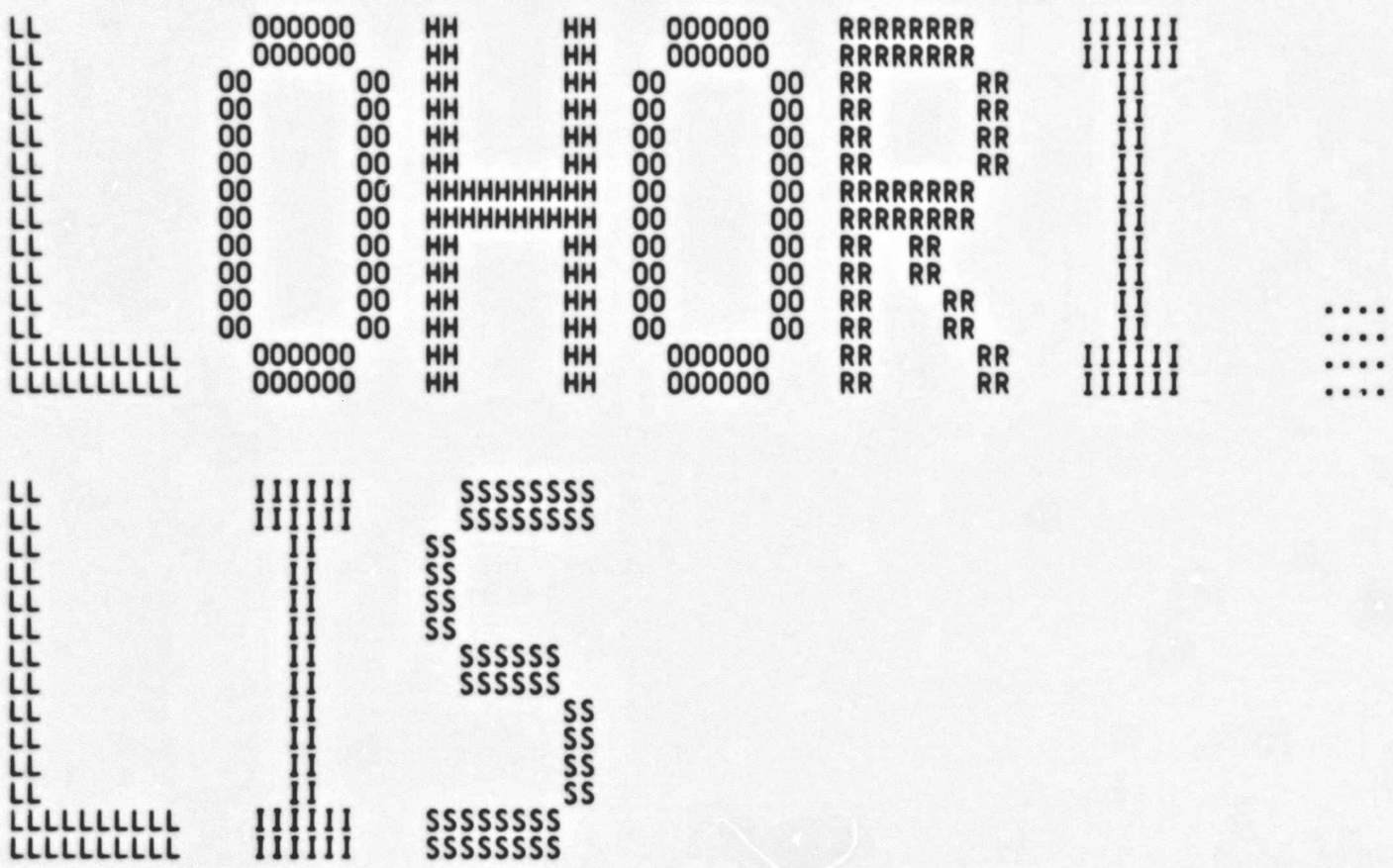
```
_$2

Sym
---
NDX
NDX
NUM
NUM
OPE
OUT
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAD
PAG
PAG
PAG
PAG
PAG
PAG
PAG
PER
PUT
RCO
RIN

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR
STR

STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

FILEID**LOHORI



LOH
V04

```
;  1          0001  0 %TITLE 'Line output (horizontal motion)'
;  2          0002  0 MODULE LOHORI ( IDENT = 'V04-000'
;  3        P 0003  0                 %BLISS32[, ADDRESSING_MODE(EXTERNAL =           LONG_RELATIVE,
;  4          0004  0                                            NONEXTERNAL =        LONG_RELATIVE)]
;  5          0005  0                 ) =
;  6          0006  1 BEGIN
;  7          0007  1 !
;  8          0008  1 !*****************************************************************************
;  9          0009  1 !*                                                                           *
; 10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 12          0012  1 !*  ALL RIGHTS RESERVED.                                                     *
; 13          0013  1 !*                                                                           *
; 14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 19          0019  1 !*  TRANSFERRED.                                                             *
; 20          0020  1 !*                                                                           *
; 21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 23          0023  1 !*  CORPORATION.                                                             *
; 24          0024  1 !*                                                                           *
; 25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 27          0027  1 !*                                                                           *
; 28          0028  1 !*                                                                           *
; 29          0029  1 !*****************************************************************************
; 30          0030  1 !
; 31          0031  1 !++
; 32          0032  1 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT: Translation from intermediate format to final output.
; 35          0035  1 !
; 36          0036  1 !
; 37          0037  1 ! ENVIRONMENT: Transportable
; 38          0038  1 !
; 39          0039  1 ! AUTHOR: K. A. Dawson   CREATION DATE: December 1983
; 40          0040  1 !
```

```
;    42          0041  1 %SBTTL 'Revision History'
;    43          0042  1 !
;    44          0043  1 ! MODIFIED BY:
;    45          0044  1 !
;    46          0045  1 !       010     KFA00010        Ken Alden       06-Jul-1983
;    47          0046  1 !               Fixed logic for resetting pointer that is used in
;    48          0047  1 !                 scanning the MRA during a no-out run-through.
;    49          0048  1 !
;    50          0049  1 !       009     KFA00009        Ken Alden       30-Jun-1983
;    51          0050  1 !               Adding a tsf_cref_count to keep track of pending crefs.
;    52          0051  1 !
;    53          0052  1 !       008     KFA00008        Ken Alden       28-Jun-1983
;    54          0053  1 !               Fixed cref bug with bars enabled.
;    55          0054  1 !
;    56          0055  1 !       007     KFA00007        Ken Alden       28-Jun-1983
;    57          0056  1 !               CLH is not called now unless line has something in it.
;    58          0057  1 !
;    59          0058  1 !       006     KFA00006        Ken Alden       27-Jun-1983
;    60          0059  1 !               Teaked the counter logic for skip_out output.
;    61          0060  1 !
;    62          0061  1 !       005     KFA00005        Ken Alden       23-Jun-1983
;    63          0062  1 !               Added call to OUTCREF when gca_skip_out was true (and a cref
;    64          0063  1 !                 is pending. This insures that the when the mra is not read,
;    65          0064  1 !                 OUTCREF may still be called.
;    66          0065  1 !
;    67          0066  1 !       004     REM00004        Ray Marshall    17-June-1983
;    68          0067  1 !               Add call to OUTCREF based on encountering a <RINTES>C escape
;    69          0068  1 !                 sequence in the MRA.
;    70          0069  1 !
;    71          0070  1 !       003     KAD00003        Keith Dawson    4-May-1983
;    72          0071  1 !               Move test for /QUICK, so that CLH is never called if we are
;    73          0072  1 !                 skipping output.
;    74          0073  1 !
;    75          0074  1 !       002     KAD00002        Keith Dawson    14-Apr-1983
;    76          0075  1 !               Correct DSRPLUS conditionals for emphasis routines.
;    77          0076  1 !
;    78          0077  1 !       001     KAD00001        Keith Dawson    22-Mar-1983
;    79          0078  1 !
;    80          0079  1 !--
```

```
;    82          0080  1 %SBTTL 'Module Level Declarations'
;    83          0081  1
;    84          0082  1 !
;    85          0083  1 ! TABLE OF CONTENTS:
;    86          0084  1 !
;    87          0085  1
;    88          0086  1 REQUIRE 'REQ:RNODEF';                   ! RUNOFF variant definitions
;    89          0217  1
;    90          0218  1 FORWARD ROUTINE
;    91          0219  1     LOUT1 : NOVALUE,
;    92          0220  1     build_line,
;    93          0221  1     compute_next_pass;
;    94          0222  1 !
;    95          0223  1 ! INCLUDE FILES:
;    96          0224  1 !
;    97          0225  1 LIBRARY 'NXPORT:XPORT';                ! XPORT Library
;    98          0226  1
;    99        U 0227  1 %IF DSRPLUS %THEN
;   100        U 0228  1 LIBRARY 'REQ:DPLLIB';                 ! DSRPLUS BLISS Library
;   101          0229  1 %ELSE
;   102          0230  1 LIBRARY 'REQ:DSRLIB';                 ! DSR BLISS Library
;   103          0231  1 %FI
;   104          0232  1 !
;   105          0233  1 ! MACROS:
;   106          0234  1 !
;   107          0235  1
;   108          0236  1 MACRO
;   109        M 0237  1     emphasis_passes =
;   110        M 0238  1         (     (.pass_cntr GTR pass_setup)
;   111        M 0239  1           AND (.pass_cntr LSS pass_real_text)
;   112        M 0240  1         )
;   113          0241  1     %;
;   114          0242  1
;   115          0243  1 MACRO
;   116        M 0244  1     doing_underlining =
;   117        M 0245  1         (.pass_cntr EQL pass_underline
;   118        M 0246  1         OR
;   119        M 0247  1          .pass_cntr EQL pass_bold_underline)
;   120          0248  1     %;
;   121          0249  1
;   122          0250  1 MACRO                                   ! TRUE if output is for an
;   123        M 0251  1     laser_output =                     ! LN01 or an LN01E.
;   124        M 0252  1         (.gca_op_dev EQL op_dev_ln01
;   125        M 0253  1         OR
;   126        M 0254  1         .gca_op_dev EQL op_dev_ln01e)
;   127          0255  1     %;
;   128          0256  1
;   129          0257  1 MACRO                                   ! TRUE if we should generate
;   130        M 0258  1     generate_bare_cr_line =     (       ! intermediate output.
;   131        M 0259  1
;   132        M 0260  1      (  (.last_pass GTR 0)              ! First BUILD_LINE loop...
;   133        M 0261  1         AND                             ! and either we are doing bolding/overstriking,
;   134        M 0262  1         ( (NOT doing_underlining)       ! or we are doing underlining but not /SEPERATE.
;   135        M 0263  1           OR
;   136        M 0264  1           (doing_underlining AND NOT .outopt_und_sep)
;   137        M 0265  1         )
;   138        M 0266  1      )
```

```
;  139          M 0267  1    OR
;  140          M 0268  1       (  (.last_pass LSS 0)                ! Second BUILD_LINE loop...
;  141          M 0269  1          AND                               ! and any bolding is present.
;  142          M 0270  1          (.tsf_bld)
;  143          M 0271  1       )
;  144          M 0272  1                                        )
;  145            0273  1       %;
;  146            0274  1 !
;  147            0275  1 ! EQUATED SYMBOLS:
;  148            0276  1 !
;  149            0277  1
;  150            0278  1 EXTERNAL LITERAL
;  151            0279  1     rintes : UNSIGNED (8);
;  152            0280  1
;  153            0281  1 !
;  154            0282  1 ! OWN STORAGE:
;  155            0283  1 !
;  156            0284  1 OWN
;  157            0285  1     emphasis_bits,                   ! A word containing information on current-character
;  158            0286  1                                      ! and last-character bold and underline.
;  159            0287  1     p_lines,                         ! Number of physical lines represented.
;  160            0288  1     padding : VECTOR [75],           ! Justification spacing built up here.
;  161            0289  1     pi,                              ! Index into padding.
;  162            0290  1     overstrike_count,                ! Number of characters in an overstrike sequence (/BACKSPACE mode only).
;  163            0291  1     overstrike_char,                 ! The character with which to overstrike the previous one.
;  164            0292  1     overstrike_seq,                  ! CH$PTR to start of an overstrike sequence.
;  165            0293  1     bold_limit,                      ! Location of last character to be bolded.
;  166            0294  1     over_limit,                      ! Location of last overstruck character.
;  167            0295  1     under_limit,                     ! Location of last character to be underlined.
;  168            0296  1     pass_limit,                      ! Limit of scan for current pass
;  169            0297  1     next_pass,                       ! Keeps track of the next pass.
;  170            0298  1     last_pass,                       ! The number of the last pass for this line.
;  171            0299  1     pass_cntr;                       ! Count of which pass is happening. See below.
;  172            0300  1
;  173            0301  1 !
;  174            0302  1 ! EXTERNAL REFERENCES:
;  175            0303  1 !
;  176            0304  1
;  177            0305  1 EXTERNAL
;  178            0306  1     fnct : FNCT_DEFINITION,
;  179          U 0307  1 %IF FLIP %THEN
;  180          U 0308  1     rnoiob: REF $XPO_IOB(),
;  181            0309  1 %FI
;  182            0310  1
;  183            0311  1     fra : fixed_string,
;  184            0312  1     gca : gca_definition,
;  185            0313  1     sca : sca_definition,
;  186            0314  1     tsf : tsf_definition,
;  187            0315  1     outopt : VECTOR [outopt_size],
;  188            0316  1     phan : phan_definition;
;  189            0317  1
;  190            0318  1 EXTERNAL LITERAL                                  ! Error messages
;  191            0319  1     rnfile;
;  192            0320  1
;  193            0321  1 EXTERNAL ROUTINE
;  194            0322  1     bsemph,     opemph,
;  195            0323  1
```

```
;  196          0324  1 %IF LN01 %THEN
;  197          0325  1     lnemph : NOVALUE,
;  198          0326  1 %FI
;  199        U 0327  1 %IF DSRPLUS %THEN
;  200        U 0328  1     vtemph : NOVALUE,   outcref,
;  201          0329  1 %FI
;  202        U 0330  1 %IF FLIP %THEN
;  203        U 0331  1     flemph : NOVALUE,
;  204          0332  1 %FI
;  205          0333  1
;  206          0334  1     clh,        cskipl,         erms,           fbwait,
;  207          0335  1     justf,      lstops,         newpag,         tpfeql,
;  208          0336  1     tpr,        uform;
```

```
;  210     0337  1 %SBTTL 'LOUT1 -- Process remaining normal text in line.'
;  211     0338  1 GLOBAL ROUTINE LOUT1 (ptr) : NOVALUE =
;  212     0339  1
;  213     0340  1 !++
;  214     0341  1 ! FUNCTIONAL DESCRIPTION:
;  215     0342  1 !
;  216     0343  1 !       Process the remaining normal text in the line.
;  217     0344  1 !
;  218     0345  1 ! FORMAL PARAMETERS:
;  219     0346  1 !
;  220     0347  1 !       ptr             Character reader in input line.
;  221     0348  1 !
;  222     0349  1 ! IMPLICIT INPUTS:      None
;  223     0350  1 !
;  224     0351  1 ! IMPLICIT OUTPUTS:     None
;  225     0352  1 !
;  226     0353  1 ! ROUTINE VALUE:
;  227     0354  1 ! COMPLETION CODES:     None
;  228     0355  1 !
;  229     0356  1 ! SIDE EFFECTS:         None
;  230     0357  1 !
;  231     0358  1 !--
;  232     0359  1
;  233     0360  2     BEGIN
;  234     0361  2
;  235     0362  2     LOCAL
;  236     0363  2         status;
;  237     0364  2
;  238     0365  2     pass_cntr = 1;
;  239     0366  2     bold_limit = 0;
;  240     0367  2     over_limit = 0;
;  241     0368  2     under_limit = 0;
;  242     0369  2     status = false;
;  243     0370  2
;  244     0371  2
;  245     0372  2     INCR i FROM 0 TO 74 DO  padding [.i] = 1;
;  246     0373  2
;  247     0374  2     ! Compute number of physical lines that this record represents.
;  248     0375  2     p_lines = 1;
;  249     0376  2
;  250     0377  3     IF (.tsf_und
;  251     0378  3         AND .outopt_und_sep)
;  252     0379  2     THEN
;  253     0380  2         p_lines = 2;            ! Underline with dashes on next line.
;  254     0381  2
;  255     0382  2     ! This 'turns a page' if necessary.
;  256     0383  2     IF NOT .fnct_expanding
;  257     0384  2     THEN
;  258     0385  3         BEGIN
;  259     0386  3         IF NOT tpr (.p_lines)
;  260     0387  3         THEN
;  261     0388  3             phan_top_page = .phan_paging OR .phan_top_page;
;  262     0389  3
;  263     0390  3         ! If we are positioned at precisely the position where it would be ok
;  264     0391  3         ! to output one or more footnotes, terminate a new page.
;  265     0392  4         IF (tpfeql () NEQ 0)
;  266     0393  3         THEN
```

```
;  267          0394  3              phan_top_page = .phan_paging OR .phan_top_page;
;  268          0395  2          END;
;  269          0396  2
;  270          0397  2      ! If necessary, put heading on page before writing text.
;  271          0398  2      IF (.phan_top_page
;  272          0399  3          AND NOT .fnct_expanding)
;  273          0400  2      THEN
;  274          0401  2          newpag ();
;  275          0402  2
;  276          0403  2      ! If skipping output because the user used the /PAGES switch (or because
;  277          0404  2      ! we are in an early pass of /AUTOMATIC processing), just count the lines
;  278          0405  2      ! but don't do any output.
;  279          0406  2      IF .gca_skip_out
;  280          0407  2      THEN
;  281          0408  3          BEGIN
;  282        U 0409  3  %IF DSRPLUS %THEN
;  283        U 0410  3          LOCAL
;  284        U 0411  3              temp_ptr,
;  285        U 0412  3              temp_length;
;  286        U 0413  3
;  287          0414  3  %FI
;  288          0415  3              phan_lines_tp = .phan_lines_tp + .p_lines;
;  289        U 0416  3  %IF DSRPLUS %THEN
;  290        U 0417  3          temp_length = .tsf_int_hl;
;  291        U 0418  3          temp_ptr = .ptr;
;  292        U 0419  3          WHILE (.tsf_cref_data NEQ 0)
;  293        U 0420  3                AND
;  294        U 0421  3           (.temp_length NEQ 0)
;  295        U 0422  3                AND
;  296        U 0423  3           (NOT CH$FAIL (.temp_ptr))      DO
;  297        U 0424  3              BEGIN
;  298        U 0425  3              temp_ptr = CH$FIND_SUB (.temp_length, .temp_ptr, 3
;  299        U 0426  3                        ,CH$PTR(%PLIT(%STRING(%CHAR (28),'C '))));
;  300        U 0427  3              IF NOT CH$FAIL (.temp_ptr)
;  301        U 0428  3              THEN
;  302        U 0429  3                  BEGIN
;  303        U 0430  3                  outcref ();     !Dump this pending cref.
;  304        U 0431  3                  tsf_cref_count = .tsf_cref_count - 1; ! One less cref pending.
;  305        U 0432  3                  !Reduce the context length to reflect what it has already scanned.
;  306        U 0433  3                  temp_length = .tsf_int_hl - CH$DIFF(.temp_ptr,.ptr) - 3;
;  307        U 0434  3                  temp_ptr = CH$PLUS(.temp_ptr, 3);
;  308        U 0435  3                  END;
;  309        U 0436  3              END;
;  310          0437  3  %FI
;  311          0438  3          RETURN
;  312          0439  2          END;
;  313          0440  2
;  314          0441  2      ! Compute spacing for justification.
;  315          0442  2      IF .tsf_jus_cnt NEQ 0                  ! If justification required,
;  316          0443  2      THEN                                   ! set up PADDING accordingly.
;  317          0444  2          justf (padding, .tsf_jus_cnt,
;  318          0445  2                         (IF .tsf_justify THEN .tsf_padding ELSE 0),
;  319          0446  2                         .tsf_just_alg);
;  320          0447  2
;  321          0448  2      ! Take care of possible pending formfeed.
;  322          0449  2      IF .phan_form_pend NEQ 0
;  323          0450  2      THEN
```

```
;  324          0451  2          IF .phan_simulate
;  325          0452  2          THEN
;  326          0453  2              uform ()                                !/SIMULATE
;  327          0454  2          ELSE
;  328          0455  2              IF .phan_pause
;  329          0456  2              THEN
;  330          0457  2                  fbwait ()
;  331          0458  2              ELSE
;  332        U 0459  2 %IF FLIP %THEN
;  333        U 0460  2                  IF NOT (.gca_op_dev EQL op_dev_flip)
;  334        U 0461  2                  THEN
;  335          0462  2 %FI
;  336          0463  2                      ! We must write out the formfeed here and then clear the
;  337          0464  2                      ! FRA, because if an emphasized title is waiting, the FRA
;  338          0465  2                      ! is going to be cleared (in BUILD_LINE) before it has
;  339          0466  2                      ! a chance to be written.
;  340          0467  3                      BEGIN
;  341          0468  3                      fs_wchar (fra, .phan_form_pend);
;  342          0469  3                      clh (clh_out_nocrlf);
;  343          0470  3                      fs_init (fra);
;  344          0471  2                      END;
;  345          0472  2
;  346          0473  2      phan_form_pend = 0;
;  347          0474  2
;  348          0475  2 !+
;  349          0476  2 ! Generation of what TSF/MRA represent happens below this point.
;  350          0477  2 !
;  351          0478  2 ! Take care of actual line printing, including bold, overstriking,
;  352          0479  2 ! and underlining.  This, if not done using backspace, requires several
;  353          0480  2 ! passes over the line to generate separate lines which can then
;  354          0481  2 ! be used to overstrike each other.
;  355          0482  2 !-
;  356          0483  2      ! Make sure the pass counter is 1 going into BUILD_LINE. This avoids a
;  357          0484  2      ! nasty bug involving recursive calls to LOUT for top-of-page processing.
;  358          0485  2      !
;  359          0486  2      pass_cntr = 1;
;  360          0487  2      WHILE NOT .status DO
;  361          0488  2          status = build_line (.ptr);
;  362          0489  2
;  363          0490  2 !Processing continues here when we exit BUILD_LINE returning TRUE.
;  364          0491  2      !1.  This output statement (also) does the last overprint
;  365          0492  2      !    to achieve the proper bolding depth.
;  366          0493  2      !2.  In /BACKSPACE mode, this write statement does the
;  367          0494  2      !    actual output, since nothing has been output yet.
;  368          0495  2      !3.  In either case, the terminating CRLF is output.
;  369        U 0496  2 %IF FLIP %THEN
;  370        U 0497  2      IF (.gca_op_dev EQL op_dev_flip) and .sca_header
;  371        U 0498  2      THEN
;  372        U 0499  2          BEGIN
;  373        U 0500  2          OWN tochl_rec : $flip_tochl PRESET ([tochl_code] = flip$k_tochl);
;  374        U 0501  2          $XPO_PUT (IOB=.rnoiob, STRING= (flip$k_tochl_size,tochl_rec));
;  375        U 0502  2          sca_header = false;
;  376        U 0503  2          END;
;  377          0504  2 %FI
;  378          0505  2      op_dev_write_output_line;
;  379          0506  2
;  380          0507  2      ! Clear output buffer for next line.
```

```
;  381          0508  2     fs_init (fra);
;  382          0509  2 !+
;  383          0510  2 ! Generate separate underlining now, if specified.
;  384          0511  2 !-
;  385          0512  2     IF  .outopt_und_sep
;  386          0513  2         AND
;  387          0514  2         .tsf_und                !Not unless there is any underlining to do!
;  388          0515  2     THEN
;  389          0516  3         BEGIN
;  390          0517  4         pass_cntr = (IF .tsf_bld
;  391          0518  4                      THEN pass_bold_underline
;  392          0519  3                      ELSE pass_underline);
;  393          0520  3         last_pass = -1;         ! "-1" signals this separate-underline call to BUILD_LINE.
;  394          0521  3         build_line (.ptr);
;  395          0522  3         fs_wchar (fra, 10);     ! Add <lf> to the line.
;  396          0523  3         clh (clh_out_nocrlf);   !
;  397          0524  3         fs_init (fra);          ! Clear output buffer for next line.
;  398          0525  3         phan_lines_tp = .phan_lines_tp + 1;
;  399          0526  2         END;
;  400          0527  2
;  401          0528  2     ! Update count of number of lines on this page.
;  402          0529  2     phan_lines_tp = .phan_lines_tp + 1;
;  403          0530  1     END;                                    ! End of LOUT1


                                                        .TITLE  LOHORI Line output (horizontal motion)
                                                        .IDENT  \V04-000\

                                                        .PSECT  $OWN$,NOEXE,2

                                                00000 EMPHASIS_BITS:
                                                                .BLKB   4
                                                00004 P_LINES:  .BLKB   4
                                                00008 PADDING:  .BLKB   300
                                                00134 PI:       .BLKB   4
                                                00138 OVERSTRIKE_COUNT:
                                                                .BLKB   4
                                                0013C OVERSTRIKE_CHAR:
                                                                .BLKB   4
                                                00140 OVERSTRIKE_SEQ:
                                                                .BLKB   4
                                                00144 BOLD_LIMIT:
                                                                .BLKB   4
                                                00148 OVER_LIMIT:
                                                                .BLKB   4
                                                0014C UNDER_LIMIT:
                                                                .BLKB   4
                                                00150 PASS_LIMIT:
                                                                .BLKB   4
                                                00154 NEXT_PASS:
                                                                .BLKB   4
                                                00158 LAST_PASS:
                                                                .BLKB   4
                                                0015C PASS_CNTR:
                                                                .BLKB   4

                                                        .EXTRN  RINTES, FNCT, FRA
```

```
                                                        .EXTRN  GCA, SCA, TSF, OUTOPT
                                                        .EXTRN  PHAN, RNFILE, BSEMPH
                                                        .EXTRN  OPEMPH, LNEMPH, CLH
                                                        .EXTRN  CSKIPL, ERMS, FBWAIT
                                                        .EXTRN  JUSTF, LSTOPS, NEWPAG
                                                        .EXTRN  TPFEQL, TPR, UFORM

                                                        .PSECT  $CODE$,NOWRT,2

                                            OFFC 00000  .ENTRY  LOUT1, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 ; 0338
                                  5B 00000000V  EF  9E 00002  MOVAB   BUILD_LINE, R11                         ;
                                  5A 00000000G  EF  9E 00009  MOVAB   FNCT+24, R10                            ;
                                  59 00000000G  EF  9E 00010  MOVAB   OUTOPT+8, R9                            ;
                                  58 00000000G  EF  9E 00017  MOVAB   CLH, R8                                 ;
                                  57 00000000G  EF  9E 0001E  MOVAB   TSF, R7                                 ;
                                  56 00000000'  EF  9E 00025  MOVAB   P_LINES, R6                             ;
                                  55 00000000G  EF  9E 0002C  MOVAB   PHAN, R5                                ;
                                  54 00000000G  EF  9E 00033  MOVAB   FRA+4, R4                               ;
                            0158  C6            01  D0 0003A  MOVL    #1, PASS_CNTR                         ; 0365
                                        0140    C6  7C 0003F  CLRQ    BOLD_LIMIT                            ; 0366
                                        0148    C6  D4 00043  CLRL    UNDER_LIMIT                           ; 0368
                                                53  D4 00047  CLRL    STATUS                                ; 0369
                                                50  D4 00049  CLRL    I                                     ; 0372
                                  04  A640          01  D0 0004B 1$:  MOVL    #1, PADDING[I]                ;
                            F3          50 0000004A 8F  F3 00050  AOBLEQ  #74, I, 1$                        ;
                                        66          01  D0 00058  MOVL    #1, P_LINES                       ; 0375
                                        50          67  D0 0005B  MOVL    TSF, R0                           ; 0377
                            06    08    A0          01  E1 0005E  BBC     #1, 8(R0), 2$                     ;
                                        03          69  E9 00063  BLBC    OUTOPT+8, 2$                      ; 0378
                                        66          02  D0 00066  MOVL    #2, P_LINES                       ; 0380
                                        1F          6A  E8 00069 2$:  BLBS    FNCT+24, 4$                   ; 0383
                                                    66  DD 0006C  PUSHL   P_LINES                           ; 0386
                  00000000G             EF          01  FB 0006E  CALLS   #1, TPR                           ;
                                        04          50  E8 00075  BLBS    R0, 3$                            ;
                                        65    28    B5  C8 00078  BISL2   @PHAN+40, PHAN                    ; 0388
                  00000000G             EF          00  FB 0007C 3$:  CALLS   #0, TPFEQL                    ; 0392
                                                    50  D5 00083  TSTL    R0                                ;
                                                    04  13 00085  BEQL    4$                                ;
                                        65    28    B5  C8 00087  BISL2   @PHAN+40, PHAN                    ; 0394
                                        0A          65  E9 0008B 4$:  BLBC    PHAN, 5$                      ; 0398
                                        07          6A  E8 0008E  BLBS    FNCT+24, 5$                       ; 0399
                  00000000G             EF          00  FB 00091  CALLS   #0, NEWPAG                        ; 0401
                                        05 00000000G  EF  E9 00098 5$:  BLBC    GCA+112, 6$                   ; 0406
                              0C        A5          66  C0 0009F  ADDL2   P_LINES, PHAN+12                  ; 0415
                                                    04  000A3    RET                                         ; 0408
                                        50          67  D0 000A4 6$:  MOVL    TSF, R0                       ; 0442
                                              20    A0  D5 000A7  TSTL    32(R0)                            ;
                                                    1B  13 000AA  BEQL    9$                                ;
                                              64    A0  DD 000AC  PUSHL   100(R0)                           ; 0446
                                        05    24    A0  E9 000AF  BLBC    36(R0), 7$                        ; 0445
                                              40    A0  DD 000B3  PUSHL   64(R0)                            ;
                                                    02  11 000B6  BRB     8$                                ;
                                                    7E  D4 000B8 7$:  CLRL    -(SP)                         ;
                                              20    A0  DD 000BA 8$:  PUSHL   32(R0)                        ; 0444
                                              04    A6  9F 000BD  PUSHAB  PADDING                           ;
                  00000000G             EF          04  FB 000C0  CALLS   #4, JUSTF                         ;
                                        52    20    A5  D0 000C7 9$:  MOVL    PHAN+32, R2                   ; 0449
```

```
                                        34  13 000CB        BEQL    12$                              :
                          09      34    A5  E9 000CD        BLBC    PHAN+52, 10$                     : 0451
               00000000G  EF            00  FB 000D1        CALLS   #0, UFORM                        : 0453
                                        27  11 000D8        BRB     12$                              :
                          09      3C    A5  E9 000DA  10$:  BLBC    PHAN+60, 11$                     : 0455
               00000000G  EF            00  FB 000DE        CALLS   #0, FBWAIT                       : 0457
                                        1A  11 000E5        BRB     12$                              :
                     00   B4            52  90 000E7  11$:  MOVB    R2, @FRA+4                       : 0468
                                        64  D6 000EB        INCL    FRA+4                            :
                                  08    A4  D6 000ED        INCL    FRA+12                           :
                                        0B  DD 000F0        PUSHL   #11                              : 0469
                          68            01  FB 000F2        CALLS   #1, CLH                          :
                                  08    A4  D4 000F5        CLRL    FRA+12                           : 0470
                     FC   A4      0C    A4  9E 000F8        MOVAB   FRA+16, FRA                      :
                          64      FC    A4  D0 000FD        MOVL    FRA, FRA+4                       :
                                  20    A5  D4 00101  12$:  CLRL    PHAN+32                          : 0473
                   0158   C6            01  D0 00104        MOVL    #1, PASS_CNTR                    : 0486
                          0B            53  E8 00109  13$:  BLBS    STATUS, 14$                      : 0487
                                  04    AC  DD 0010C        PUSHL   PTR                              : 0488
                          6B            01  FB 0010F        CALLS   #1, BUILD_LINE                   :
                          53            50  D0 00112        MOVL    R0, STATUS                       :
                                        F2  11 00115        BRB     13$                              :
    02 00000000G  EF              04    04  ED 00117  14$:  CMPZV   #4, #4, GCA+208, #2              : 0503
                                        04  14 00120        BGTR    15$                              :
                                        06  DD 00122        PUSHL   #6                               :
                                        02  11 00124        BRB     16$                              :
                                        0B  DD 00126  15$:  PUSHL   #11                              :
                          68            01  FB 00128  16$:  CALLS   #1, CLH                          :
                                  08    A4  D4 0012B        CLRL    FRA+12                           : 0508
                     FC   A4      0C    A4  9E 0012E        MOVAB   FRA+16, FRA                      :
                          64      FC    A4  D0 00133        MOVL    FRA, FRA+4                       :
                          44            69  E9 0C137        BLBC    OUTOPT+8, 19$                    : 0512
                          50            67  D0 0013A        MOVL    TSF, R0                          : 0514
              3C     08   A0            01  E1 0013D        BBC     #1, 8(R0), 19$                   :
                          50            67  D0 00142        MOVL    TSF, R0                          : 0517
                          05      08    A0  E9 00145        BLBC    8(R0), 17$                       :
                          50            04  D0 00149        MOVL    #4, R0                           :
                                        03  11 0014C        BRB     18$                              :
                          50            03  D0 0014E  17$:  MOVL    #3, R0                           :
                   0158   C6            50  D0 00151  18$:  MOVL    R0, PASS_CNTR                    :
                   0154   C6            01  CE 00156        MNEGL   #1, LAST_PASS                    : 0520
                                  04    AC  DD 0015B        PUSHL   PTR                              : 0521
                          6B            01  FB 0015E        CALLS   #1, BUILD_LINE                   :
                     00   B4            0A  90 00161        MOVB    #10, @FRA+4                      : 0522
                                        64  D6 00165        INCL    FRA+4                            :
                                  08    A4  D6 00167        INCL    FRA+12                           :
                                        0B  DD 0016A        PUSHL   #11                              : 0523
                          68            01  FB 0016C        CALLS   #1, CLH                          :
                                  08    A4  D4 0016F        CLRL    FRA+12                           : 0524
                     FC   A4      0C    A4  9E 00172        MOVAB   FRA+16, FRA                      :
                          64      FC    A4  D0 00177        MOVL    FRA, FRA+4                       :
                                  0C    A5  D6 0017B        INCL    PHAN+12                          : 0525
                                  0C    A5  D6 0017E  19$:  INCL    PHAN+12                          : 0529
                                            04 00181        RET                                      : 0530

; Routine Size:  386 bytes,     Routine Base:  $CODE$ + 0000
```

```
;  405     0531  1 %SBTTL 'BUILD_LINE -- output entire text line, using multiple passes if needed'
;  406     0532  1 ROUTINE BUILD_LINE (ptr) =
;  407     0533  1
;  408     0534  1 !++
;  409     0535  1 ! FUNCTIONAL DESCRIPTION:
;  410     0536  1 !
;  411     0537  1 !       BUILD_LINE does the actual work of building up an output line,
;  412     0538  1 !       using multiple passes for overstriking, underlining, and bolding.
;  413     0539  1 !
;  414     0540  1 ! FORMAL PARAMETERS:
;  415     0541  1 !
;  416     0542  1 !       ptr             is the input line pointer, passed from LOUT to LOUT1.
;  417     0543  1 !
;  418     0544  1 ! IMPLICIT INPUTS:
;  419     0545  1 !
;  420     0546  1 !       Some of the OWN variables of this module (NOUT).
;  421     0547  1 !
;  422     0548  1 ! IMPLICIT OUTPUTS:     None
;  423     0549  1 !
;  424     0550  1 ! ROUTINE VALUE:
;  425     0551  1 ! COMPLETION CODES:
;  426     0552  1 !
;  427     0553  1 !       The routine returns TRUE to indicate that more processing is
;  428     0554  1 !       required on a line; it returns FALSE to indicate that the line is
;  429     0555  1 !       ready for output. It is called in a loop until it returns FALSE.
;  430     0556  1 !
;  431     0557  1 ! SIDE EFFECTS:
;  432     0558  1 !
;  433     0559  1 !       Text is written onto FRA.
;  434     0560  1 !--
;  435     0561  1
;  436     0562  2     BEGIN
;  437     0563  2
;  438     0564  2     LOCAL
;  439     0565  2         hold_khar,
;  440     0566  2         hold_operand1,
;  441     0567  2         hold_seq_start,         ! CH$PTR to start of a character sequence.
;  442     0568  2         op_code,
;  443     0569  2         operand1,
;  444     0570  2         ptr_copy_1;
;  445     0571  2
;  446     0572  2     ! Initialize LOCAL variables.
;  447     0573  2     !
;  448     0574  2     hold_operand1 = 0;
;  449     0575  2     operand1 = 0;
;  450     0576  2     ptr_copy_1 = .ptr;
;  451     0577  2     next_pass = .pass_cntr;     ! Initialize to the current pass count.
;  452     0578  2
;  453     0579  2     ! Initialize OWN variables.
;  454     0580  2     !
;  455     0581  2     pi = 0;
;  456     0582  2
;  457     0583  2     ! Initialize emphasis and overstriking information for each call.
;  458     0584  2     !
;  459     0585  2     emphasis_bits = 0;
;  460     0586  2     overstrike_count = 0;
;  461     0587  2     overstrike_char = 0;
```

```
;  462     0588  2
;  463     0589  2      ! For LN01[e] output, overstriking is treated in a special way. In this
;  464     0590  2      ! case, bolding/underlining passes produce no text on the FRA; the only
;  465     0591  2      ! passes that write to the FRA are pass_overstrike and pass_real_text.
;  466     0592  2      !
;  467     0593  3      IF  (laser_output AND .tsf_ovr)
;  468     0594  3          AND
;  469     0595  3          (.pass_cntr LSS pass_overstrike)
;  470     0596  3          AND
;  471     0597  3          (.pass_cntr GTR pass_setup)
;  472     0598  2      THEN
;  473     0599  2          ! Decide which pass comes next and set up the counter for it.
;  474     0600  3          BEGIN
;  475     0601  3          pass_cntr = compute_next_pass ();
;  476     0602  3          RETURN false;
;  477     0603  2          END;
;  478     0604  2
;  479     0605  2      !+
;  480     0606  2      ! For all passes except text-generating one, output spaces instead
;  481     0607  2      ! of the change bars.
;  482     0608  2      !-
;  483     0609  3      IF emphasis_passes
;  484     0610  2      THEN
;  485     0611  2          lstops (lstops_none, false)     ! Space over the listing option columns.
;  486     0612  2      ELSE
;  487     0613  2          lstops (lstops_all, false);     ! Output listing options.
;  488     0614  2
;  489     0615  2      ! Shift text according to amount computed by .CENTER, etc commands.
;  490     0616  2
;  491     0617  2      INCR i FROM 1 TO .tsf_adjust DO
;  492     0618  2          fs_wchar (fra, %C' ');
;  493     0619  2
;  494     0620  2      ! Get limit of scan for this pass.
;  495     0621  3      pass_limit = (CASE .pass_cntr FROM pass_setup TO pass_real_text OF
;  496     0622  3          SET
;  497     0623  3
;  498     0624  3          [pass_setup] :           .tsf_int_hl;   ! 1st pass sets up others.
;  499     0625  3
;  500     0626  3          [pass_bold] :            .bold_limit;   ! Last character for bolding
;  501     0627  3
;  502     0628  3          [pass_overstrike] :      .over_limit;   ! Last overstriking character
;  503     0629  3          [pass_bold_overstrike] : .over_limit;
;  504     0630  3
;  505     0631  3          [pass_underline] :       .under_limit;  ! Last underlined character
;  506     0632  3          [pass_bold_underline] :  .under_limit;
;  507     0633  3
;  508     0634  3          [pass_real_text] :       .tsf_int_hl;   ! 7th pass generates output
;  509     0635  3
;  510     0636  2          TES);
;  511     0637  2
;  512     0638  2      INCR k FROM 1 TO .pass_limit DO     ! Process (horizontal) text.
;  513     0639  3          BEGIN
;  514     0640  3          hold_seq_start = .ptr_copy_1;   ! Remember start of this sequence.
;  515     0641  3          hold_khar = CH$RCHAR_A (ptr_copy_1);
;  516     0642  3
;  517     0643  3          IF .hold_khar EQL rintes
;  518     0644  3          THEN
```

```
  4                 BEGIN
  4                 op_code = CH$RCHAR_A (ptr_copy_1);
  4                 hold_operand1 = CH$RCHAR_A (ptr_copy_1);
  4                 k = .k + 2;
  4
  4                 SELECT .op_code OF
  4                     SET
  4
  4                     [%C'B'] :
  5                         (IF .tsf_bld
  5                         THEN
  5                         ! Remember bolding information if bolding wanted.
  6                             BEGIN
  6                             emph_current_bold = true;
  6                             operand1 = .hold_operand1;
  6                             END
  4                         );
  4
U 4 %IF DSRPLUS %THEN
U 4                 [%C'C'] :
U 4                     BEGIN
U 4                     IF .pass_cntr EQL pass_setup
U 4                     THEN
U 4                         BEGIN
U 4                         outcref();  ! Process pending .REF records.
U 4                         tsf_cref_count = .tsf_cref_count - 1;
U 4                         END;
U 4                     operand1 = 0;
U 4                     END;
  4 %FI
  4                     [%C'U'] :
  5                         (IF .tsf_und
  5                         THEN
  5                         ! Remember underlining information if underlining wanted.
  6                             BEGIN
  6                             emph_current_underline = true;
  6                             operand1 = .hold_operand1;
  6                             END
  4                         );
  4
  4                     [%C'N'] :
  4                         ! A No-operation
  5                         BEGIN
  5                         0                               ! Avoid compiler message
  4                         END;
  4
  4                     [%C'I'] :
  4                         ! Insert this character.
  5                         BEGIN
  5                         fs_wchar (fra, .hold_operand1);
  4                         END;
  4
  4                     [%C'J'] :
  4                         ! Justification mark
  5                         BEGIN
  5                         ! Insert appropriate amount of spacing here.
  5
```

```
;  576     0702  5                    INCR i FROM 1 TO .padding [.pi] DO
;  577     0703  5                        fs_wchar (fra, %C' ');
;  578     0704  5
;  579     0705  5                    operand1 = 0;
;  580     0706  5                    pi = .pi + 1;          ! Sychronize insert count with word
;  581     0707  4                    END;                   ! count.
;  582     0708  4
;  583     0709  4                [%C'O'] :
;  584     0710  5                    (IF .tsf_ovr
;  585     0711  5                    THEN
;  586     0712  5                    ! If overstriking is wanted remember this information.
;  587     0713  6                        BEGIN
;  588     0714  6                        ! Remember overstrike character
;  589     0715  6                        overstrike_char = .hold_operand1;
;  590     0716  6                        operand1 = .hold_operand1;
;  591     0717  6
;  592     0718  6                        IF .overstrike_count EQL 0
;  593     0719  6                        THEN
;  594     0720  6                        ! Remember start of overstrike sequence.
;  595     0721  6                            overstrike_seq = .hold_seq_start;
;  596     0722  6                        overstrike_count = .overstrike_count + 1;
;  597     0723  6                        END
;  598     0724  4                    );
;  599     0725  4
;  600     0726  4                [OTHERWISE] :
;  601     0727  5                    BEGIN
;  602     0728  5                    ! Some illegal character following RINTES.  Tell the user
;  603     0729  5                    ! it's an internal logic error and then carry on.
;  604     0730  5                    erms (rnfile, CH$PTR (UPLIT ('lout1')), 5);
;  605     0731  4                    END;
;  606     0732  4                TES;
;  607     0733  4            END
;  608     0734  3        ELSE
;  609     0735  4            BEGIN                  ! Are positioned at the "naked" character.
;  610     0736  4            ! Is this an emphasized or overstruck character?
;  611     0737  4            IF .operand1 NEQ 0
;  612     0738  4            THEN
;  613     0739  4                ! Process character according to which pass.
;  614     0740  5                BEGIN
;  615     0741  5                IF .pass_cntr EQL pass_setup
;  616     0742  5                THEN
;  617     0743  5                    ! Save location of emphasized character for later passes.
;  618     0744  6                    BEGIN
;  619     0745  6                    IF .emph_current_bold                THEN bold_limit  = .k;
;  620     0746  6                    IF (.overstrike_count NEQ 0)         THEN over_limit  = .k;
;  621     0747  6                    IF .emph_current_underline           THEN under_limit = .k;
;  622     0748  5                    END;
;  623     0749  5
;  624     0750  5                SELECTONE TRUE OF
;  625     0751  5                    SET
;  626     0752  5 %IF LN01 %THEN
;  627     0753  5                    [laser_output] :             !Output for LN01 or LN01E.
;  628     0754  5                        lnemph (.hold_khar, .gca_ln01_ital_under,
;  629     0755  5                                emphasis_bits, .overstrike_count,
;  630     0756  5                                .overstrike_char, .overstrike_seq, .pass_cntr);
;  631     0757  5 %FI
;  632   U 0758  5 %IF DSRPLUS %THEN
```

```
;  633     U 0759  5                       [.gca_op_dev EQL op_dev_vt100] :    !Output for VT100.
;  634     U 0760  5                           vtemph (.hold_khar, .gca_ln01_ital_under,
;  635     U 0761  5                                   emphasis_bits, .pass_cntr);
;  636       0762  5 %FI
;  637     U 0763  5 %IF FLIP %THEN
;  638     U 0764  5                       [(.gca_op_dev EQL op_dev_flip)] :    !Output for FLIP.
;  639     U 0765  5                           flemph (.hold_khar, .gca_ln01_ital_under,
;  640     U 0766  5                                   emphasis_bits, .pass_cntr);
;  641       0767  5 %FI
;  642       0768  5                       [.outopt_back] :                    !Backspace mode.
;  643       0769  5                           bsemph (.hold_khar, .gca_ln01_ital_under,
;  644       0770  5                                   emphasis_bits, .overstrike_count,
;  645       0771  5                                   .overstrike_char, .overstrike_seq, .pass_cntr);
;  646       0772  5
;  647       0773  5                       [.outopt_over] :                    !Line overprinting mode.
;  648       0774  5                           opemph  (.hold_khar, .gca_ln01_ital_under,
;  649       0775  5                                   emphasis_bits, .overstrike_count,
;  650       0776  5                                   .overstrike_char, .overstrike_seq, .pass_cntr);
;  651       0777  5
;  652       0778  5                       [OTHERWISE] :
;  653       0779  5                           erms (rnfile, CH$PTR (UPLIT ('build_line')), 10);
;  654       0780  5
;  655       0781  5                       TES;
;  656       0782  5
;  657       0783  5                   operand1 = 0;
;  658       0784  5                   hold_operand1 = 0;
;  659       0785  5                   emph_current_bold = false;
;  660       0786  5                   emph_current_underline = false;
;  661       0787  5                   overstrike_count = 0;
;  662       0788  5                   overstrike_char = 0;
;  663       0789  5                   END
;  664       0790  4               ELSE
;  665       0791  4                   !It's a normal, unemphasized character to be output. Put it
;  666       0792  4                   ! in the output buffer only if pass 1 or pass 7; otherwise
;  667       0793  4                   ! use ' ' for a place holder.
;  668       0794  5                   IF (NOT emphasis_passes)
;  669       0795  4                   THEN
;  670       0796  5                       BEGIN
;  671       0797  5 %IF LN01 %THEN
;  672       0798  5                       !Check for LN01 emphasis and turn it off.
;  673       0799  6                       IF laser_output
;  674       0800  5                           AND
;  675       0801  6                           (.emph_previous_emphasized NEQ 0)       !Emphasis on?
;  676       0802  5                       THEN
;  677       0803  5                           !Have to turn off all emphasis
;  678       0804  5                           lnemph (-1, .gca_ln01_ital_under,
;  679       0805  5                                   emphasis_bits, .overstrike_count,
;  680       0806  5                                   .overstrike_char, .overstrike_seq, .pass_cntr);
;  681       0807  5 %FI
;  682     U 0808  5 %IF DSRPLUS %THEN
;  683     U 0809  5                       !Check for VT100 emphasis and turn it off.
;  684     U 0810  5                       IF (.gca_op_dev EQL op_dev_vt100)           !VT100.
;  685     U 0811  5                           AND
;  686     U 0812  5                           (.emph_previous_emphasized NEQ 0)       !Emphasis on?
;  687     U 0813  5                       THEN
;  688     U 0814  5                           !Have to turn off all emphasis
;  689     U 0815  5                           vtemph (-1, .gca_ln01_ital_under,
```

```
;  690        U 0816  5                                   emphasis_bits, .pass_cntr);
;  691          0817  5 %FI
;  692        U 0818  5 %IF FLIP %THEN
;  693        U 0819  5                       !Check for FLIP emphasis and turn it off
;  694        U 0820  5                       IF (.gca_op_dev EQL op_dev_flip)
;  695        U 0821  5                       THEN
;  696        U 0822  5                           flemph (-1, .gca_ln01_ital_under,
;  697        U 0823  5                                   emphasis_bits, .pass_cntr);
;  698          0824  5 %FI
;  699          0825  5                       fs_wchar (fra, .hold_khar); !First or last pass: write character.
;  700          0826  5                       END
;  701          0827  4                   ELSE
;  702          0828  4                       fs_wchar (FRA, %C' ');      ! Emphasis pass: write placeholding space.
;  703          0829  4
;  704          0830  3               END;                                ! End of "naked" character processing.
;  705          0831  3
;  706          0832  2           END;                                    ! End of 'INCR K' loop.
;  707          0833  2
;  708          0834  2       ! For the first pass, compute the number of the LAST pass that will be
;  709          0835  2       ! made over this text.
;  710          0836  2
;  711          0837  2       IF .pass_cntr EQL pass_setup
;  712          0838  2       THEN
;  713          0839  2           last_pass =
;  714          0840  4               (IF (.over_limit NEQ 0
;  715          0841  4                    OR .under_limit NEQ 0
;  716          0842  4                    OR .bold_limit NEQ 0)
;  717          0843  3                THEN
;  718          0844  3                   pass_real_text
;  719          0845  3                ELSE
;  720          0846  2                   pass_setup);
;  721          0847  2 %IF LN01 %THEN
;  722          0848  2       ! Be sure there's no emphasis left hanging around.
;  723          0849  3       IF (laser_output AND (.emph_previous_emphasized NEQ 0) )
;  724          0850  2       THEN
;  725          0851  2           !Finish this record.
;  726          0852  2           lnemph (-1, .gca_ln01_ital_under,
;  727          0853  2                   emphasis_bits, .overstrike_count,
;  728          0854  2                   .overstrike_char, .overstrike_seq, .pass_cntr);
;  729          0855  2 %FI
;  730        U 0856  2 %IF DSRPLUS %THEN
;  731        U 0857  2       !In VT100 mode everything is done in one pass. But first we have to be
;  732        U 0858  2       !sure there's no emphasis left hanging around.
;  733        U 0859  2
;  734        U 0860  2       IF (.gca_op_dev EQL op_dev_vt100)           !VT100?
;  735        U 0861  2       THEN
;  736        U 0862  2           !Finish this record
;  737        U 0863  2           BEGIN
;  738        U 0864  2           IF .emph_previous_underline OR .emph_previous_bold      !Any emphasis left on?
;  739        U 0865  2           THEN
;  740        U 0866  2               vtemph (-1, .gca_ln01_ital_under,
;  741        U 0867  2                   emphasis_bits, .pass_cntr);
;  742        U 0868  2
;  743        U 0869  2           RETURN true;                            !Nothing more left to do.
;  744        U 0870  2           END;
;  745          0871  2 %FI
;  746        U 0872  2 %IF FLIP %THEN
```

```
;  747     U 0873  2      !For FLIP, everything is done in one pass.
;  748     U 0874  2      IF (.gca_op_dev EQL op_dev_flip)
;  749     U 0875  2      THEN
;  750     U 0876  2          BEGIN
;  751     U 0877  2          flemph (-1, .gca_ln01_ital_under,
;  752     U 0878  2                  emphasis_bits, .pass_cntr);
;  753     U 0879  2
;  754     U 0880  2          RETURN true;
;  755     U 0881  2          END;
;  756       0882  2 %FI
;  757       0883  2
;  758       0884  2 !+
;  759       0885  2 ! We are now finished with one pass ... almost. Regardless of how we got here, there's some text
;  760       0886  2 ! that hasn't been output yet. If there is nothing left to do, then we'll RETURN TRUE, which forces
;  761       0887  2 ! out this text with a CRLF after it at the bottom of LOUT1. Otherwise, near the end of this routine
;  762       0888  2 ! the text gets output with no CRLF.
;  763       0889  2 !-
;  764       0890  3      IF (.outopt_back)                    ! User said /BACKSPACE; everything is done in one pass.
;  765       0891  2        OR
;  766       0892  3          (.pass_cntr EQL .last_pass)      ! Last pass.
;  767       0893  2      THEN
;  768       0894  2          RETURN true;
;  769       0895  2
;  770       0896  2      ! For emphasis passes, add a "bare <cr>" to the line. It is at the end of the line which is to
;  771       0897  2      ! be overprinted, not at the start of the line which does the overprinting.
;  772       0898  3      IF  emphasis_passes
;  773       0899  2      THEN
;  774       0900  2          fs_wchar (fra, 13);
;  775       0901  2 !+
;  776       0902  2 ! The following two output operations, which generate the intermediate (emphasis) bare-<cr>
;  777       0903  2 ! lines, are not done for the underlining passes if the user specified /SEPARATE. Instead,
;  778       0904  2 ! underlining is done in a separate call to BUILD_LINE, from LOUT1, after the 7th pass over
;  779       0905  2 ! the line. This second call to BUILD_LINE is signalled by a LAST_PASS value of -1, which is
;  780       0906  2 ! used in the macro GENERATE_BARE_CR_LINE, tested below.
;  781       0907  2 !-
;  782       0908  2      ! If this is a bolding pass (.PASS_CNTR is even) then repeat the line as many times as specified
;  783       0909  2      ! on the /BOLD:n switch. The expression (.OUTOPT_BLDN - 1) is arrived at as follows: .OUTOPT_BLDN is
;  784       0910  2      ! the number of times that the line should be overprinted, so this INCR overprints one time less. An
;  785       0911  2      ! additional CLH (CLH_OUT_NOCRLF) below adds an overprint. When BUILD_LINE returns TRUE the final
;  786       0912  2      ! overprinting is done. The module DOOPTS has taken care of the /BOLD:0 case, so that if the user said
;  787       0913  2      ! /BOLD:0, no bolding is seen by this routine at all.
;  788       0914  2      !
;  789       0915  3      IF  (NOT .pass_cntr)
;  790       0916  2          AND
;  791       0917  3          generate_bare_cr_line
;  792       0918  2          AND
;  793       0919  3          (NOT laser_output)
;  794       0920  2      THEN
;  795       0921  2          INCR I FROM 1 TO (.outopt_bldn - 1) DO
;  796       0922  2              clh (clh_out_nocrlf);
;  797       0923  2
;  798       0924  2      ! At the end of every intermediate (emphasis) pass over the line, output the line without any
;  799       0925  2      ! carriage control following it. If this is the first or the last pass, then we do no output now.
;  800       0926  3      IF  emphasis_passes
;  801       0927  2          AND
;  802       0928  3          generate_bare_cr_line
;  803       0929  2          AND
```

```
;  804     0930  3          (NOT laser_output)
;  805     0931  2      THEN
;  806     0932  2          clh (clh_out_nocrlf);
;  807     0933  2
;  808     0934  2      !
;  809     0935  2      ! Clear the line buffer in the following cases:
;  810     0936  2      !
;  811     0937  2      !   - for LN01 output, on any pass but overstriking or real_text
;  812     0938  2      !   - for non-LN01 output, on any pass but the separate-underlining pass
;  813     0939  2      !
;  814     0940  2      IF laser_output
;  815     0941  2      THEN
;  816     0942  2          ! ** NOTE: This test depends on the values in PASS.REQ: the
;  817     0943  2          !          overstriking passes must come last.
;  818     0944  3          BEGIN
;  819     0945  3          IF .pass_cntr LSS pass_overstrike
;  820     0946  3          THEN
;  821     0947  3              fs_init (fra);
;  822     0948  3          END
;  823     0949  2      ELSE
;  824     0950  3          BEGIN
;  825     0951  3          IF .last_pass GTR 0
;  826     0952  3          THEN                                !Not separate-underlining.
;  827     0953  3              fs_init (fra);
;  828     0954  2          END;
;  829     0955  2
;  830     0956  2      ! Decide which pass comes next and set up counter for the next iteration.
;  831     0957  2      !
;  832     0958  2      pass_cntr = compute_next_pass ();
;  833     0959  2
;  834     0960  2      RETURN false;
;  835     0961  2
;  836     0962  1      END;                                        ! End of build_line


                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                        00 00 00 31 74 75 6F 6C 00000 P.AAA:    .ASCII  \lout1\<0><0><0>
00 00 65 6E 69 6C 5F 64 6C 69 75 62 00008 P.AAB:    .ASCII  \build_line\<0><0>                  ;


                                                    .PSECT  $CODE$,NOWRT,2

                                        0FFC 00000 BUILD_LINE:
                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0532
                     5B 00000000'    EF  9E 00002   MOVAB   EMPHASIS_BITS, R11                      ;
                                     56  7C 00009   CLRQ    OPERAND1                                ; 0575
                     58       04     AC  D0 0000B   MOVL    PTR, PTR_COPY_1                         ; 0576
                     51     015C     CB  D0 0000F   MOVL    PASS_CNTR, R1                           ; 0577
                0154 CB               51  D0 00014   MOVL    R1, NEXT_PASS                           ;
                                      6B  D4 00019   CLRL    EMPHASIS_BITS                           ; 0585
                           0134       CB  7C 0001B   CLRQ    PI                                      ; 0581
                           013C       CB  D4 0001F   CLRL    OVERSTRIKE_CHAR                         ; 0587
04 00000000G   EF          04         04  ED 00023   CMPZV   #4, #4, GCA+208, #4                     ; 0593
                                      0B  13 0002C   BEQL    1$                                      ;
```

```
     05 00000000G  EF          04            04  ED 0002E          CMPZV    #4, #4, GCA+208, #5              ;
                                             19  12 00037          BNEQ     2$                               ;
                               50 00000000G  EF  D0 00039 1$:      MOVL     TSF, R0                          ;
                  0D     08    A0            02  E1 00040          BBC      #2, 8(R0), 2$                    ;
                               05            51  D1 00045          CMPL     R1, #5                           ; 0595
                                             08  18 00048          BGEQ     2$                               ;
                               01            51  D1 0004A          CMPL     R1, #1                           ; 0597
                                             03  15 0004D          BLEQ     2$                               ;
                                           0469  31 0004F          BRW      69$                              ;
                               01      015C  CB  D1 00052 2$:      CMPL     PASS_CNTR, #1                    ; 0609
                                             0B  15 00057          BLEQ     3$                               ;
                               07      015C  CB  D1 00059          CMPL     PASS_CNTR, #7                    ;
                                             04  18 0005E          BGEQ     3$                               ;
                                             7E  7C 00060          CLRQ     -(SP)                            ; 0611
                                             03  11 00062          BRB      4$                               ;
                               7E            1F  7D 00064 3$:      MOVQ     #31, -(SP)                       ; 0613
                   00000000G   EF            02  FB 00067 4$:      CALLS    #2, LSTOPS                       ;
                               50 00000000G  EF  D0 0006E          MOVL     TSF, R0                          ; 0617
                                             51  D4 00075          CLRL     I                                ;
                                             13  11 00077          BRB      6$                               ;
                   00000000G   FF            20  90 00079 5$:      MOVB     #32, @FRA+4                      ; 0618
                                  00000000G  EF  D6 00080          INCL     FRA+4                            ;
                                  00000000G  EF  D6 00086          INCL     FRA+12                           ;
                        E8     51        28  A0  F3 0008C 6$:      AOBLEQ   40(R0), I, 5$                    ; 0617
                        06     01      015C  CB  CF 00091          CASEL    PASS_CNTR, #1, #6                ; 0621
        001E          001E   0010          0025     00097 7$:      .WORD    11$-7$,-                         ;
                      0025   0017          0017     0009F                   8$-7$,-                          ;
                                                                            10$-7$,-                         ;
                                                                            10$-7$,-                         ;
                                                                            9$-7$,-                          ;
                                                                            9$-7$,-                          ;
                                                                            11$-7$                           ;
                                             15  11 000A5          BRB      11$                              ; 0624
                               50      0144  CB  D0 000A7 8$:      MOVL     BOLD_LIMIT, R0                   ; 0626
                                             11  11 000AC          BRB      12$                              ;
                               50      0148  CB  D0 000AE 9$:      MOVL     OVER_LIMIT, R0                   ; 0629
                                             0A  11 000B3          BRB      12$                              ;
                               50      014C  CB  D0 000B5 10$:     MOVL     UNDER_LIMIT, R0                  ; 0632
                                             03  11 000BA          BRB      12$                              ;
                               50            60  D0 000BC 11$:     MOVL     (R0), R0                         ; 0634
                        0150   CB            50  D0 000BF 12$:     MOVL     R0, PASS_LIMIT                   ; 0621
                               59      0150  CB  D0 000C4          MOVL     PASS_LIMIT, R9                   ; 0638
                                             53  D4 000C9          CLRL     K                                ;
                                           0244  31 000CB          BRW      42$                              ;
                               5A            58  D0 000CE 13$:     MOVL     PTR_COPY_1, HOLD_SEQ_START       ; 0640
                               55            88  9A 000D1          MOVZBL   (PTR_COPY_1)+, HOLD_KHAR         ; 0641
                   00000000G   8F            55  D1 000D4          CMPL     HOLD_KHAR, #RINTES               ; 0643
                                             03  13 000DB          BEQL     14$                              ;
                                           00E8  31 000DD          BRW      25$                              ;
                               54            88  9A 000E0 14$:     MOVZBL   (PTR_COPY_1)+, OP_CODE           ; 0646
                               57            88  9A 000E3          MOVZBL   (PTR_COPY_1)+, HOLD_OPERAND1     ; 0647
                               53            02  C0 000E6          ADDL2    #2, K                            ; 0648
                               51            01  D0 000E9          MOVL     #1, R1                           ; 0650
                   00000042    8F            54  D1 000EC          CMPL     OP_CODE, #66                     ; 0653
                                             13  12 000F3          BNEQ     15$                              ;
                                             51  D4 000F5          CLRL     R1                               ;
                               50 00000000G  EF  D0 000F7          MOVL     TSF, R0                          ; 0654
```

```
                        06        08  A0  E9 000FE       BLBC    8(R0), 15$                          ;
                        6B            02  88 00102       BISB2   #2, EMPHASIS_BITS                   ; 0658
                        56            57  D0 00105       MOVL    HOLD_OPERAND1, OPERAND1             ; 0659
              00000055  8F            54  D1 00108 15$:  CMPL    OP_CODE, #85                        ; 0675
                                      14  12 0010F       BNEQ    16$                                 ;
                                      51  D4 00111       CLRL    R1                                  ;
                        50 00000000G  EF  D0 00113       MOVL    TSF, R0                             ; 0676
         06        08   A0            01  E1 0011A       BBC     #1, 8(R0), 16$                      ;
                        6B            04  88 0011F       BISB2   #4, EMPHASIS_BITS                   ; 0680
                        56            57  D0 00122       MOVL    HOLD_OPERAND1, OPERAND1             ; 0681
              0000004E  8F            54  D1 00125 16$:  CMPL    OP_CODE, #78                        ; 0685
                                      02  12 0012C       BNEQ    17$                                 ;
                                      51  D4 0012E       CLRL    R1                                  ;
              00000049  8F            54  D1 00130 17$:  CMPL    OP_CODE, #73                        ; 0691
                                      15  12 00137       BNEQ    18$                                 ;
                                      51  D4 00139       CLRL    R1                                  ;
              00000000G FF            57  90 0013B       MOVB    HOLD_OPERAND1, @FRA+4               ; 0694
                           00000000G  EF  D6 00142       INCL    FRA+4                               ;
                           00000000G  EF  D6 00148       INCL    FRA+12                              ;
              0000004A  8F            54  D1 0014E 18$:  CMPL    OP_CODE, #74                        ; 0697
                                      28  12 00155       BNEQ    21$                                 ;
                        50      0134  CB  D0 00157       MOVL    PI, R0                              ; 0702
                                      51  7C 0015C       CLRQ    R1                                  ; 0697
                                      13  11 0015E       BRB     20$                                 ; 0702
              00000000G FF            20  90 00160 19$:  MOVB    #32, @FRA+4                         ; 0703
                           00000000G  EF  D6 00167       INCL    FRA+4                               ;
                           00000000G  EF  D6 0016D       INCL    FRA+12                              ;
         E7             52        08 AB40 F3 00173 20$:  AOBLEQ  PADDING[R0], I, 19$                 ; 0702
                                      56  D4 00179       CLRL    OPERAND1                            ; 0705
                                0134  CB  D6 0017B       INCL    PI                                  ; 0706
              0000004F  8F            54  D1 0017F 21$:  CMPL    OP_CODE, #79                        ; 0709
                                      25  12 00186       BNEQ    23$                                 ;
                                      51  D4 00188       CLRL    R1                                  ;
                        50 00000000G  EF  D0 0018A       MOVL    TSF, R0                             ; 0710
         17        08   A0            02  E1 00191       BBC     #2, 8(R0), 23$                      ;
                  013C  CB            57  D0 00196       MOVL    HOLD_OPERAND1, OVERSTRIKE_CHAR      ; 0715
                        56            57  D0 0019B       MOVL    HOLD_OPERAND1, OPERAND1             ; 0716
                                0138  CB  D5 0019E       TSTL    OVERSTRIKE_COUNT                    ; 0718
                                      05  12 001A2       BNEQ    22$                                 ;
                  0140  CB            5A  D0 001A4       MOVL    HOLD_SEQ_START, OVERSTRIKE_SEQ      ; 0721
                                0138  CB  D6 001A9 22$:  INCL    OVERSTRIKE_COUNT                    ; 0722
                        15            51  E9 001AD 23$:  BLBC    R1, 24$                             ; 0726
                                      05  DD 001B0       PUSHL   #5                                  ; 0730
                           00000000'  EF  9F 001B2       PUSHAB  P.AAA                               ;
                           00000000G  8F  DD 001B8       PUSHL   #RNFILE                             ;
              00000000G EF            03  FB 001BE       CALLS   #3, ERMS                            ;
                                    014A  31 001C5 24$:  BRW     42$                                 ; 0643
                        51      015C  CB  D0 001C8 25$:  MOVL    PASS_CNTR, R1                       ; 0741
                                      56  D5 001CD       TSTL    OPERAND1                            ; 0737
                                      03  12 001CF       BNEQ    26$                                 ;
                                    00DD  31 001D1       BRW     36$                                 ;
                        01            51  D1 001D4 26$:  CMPL    R1, #1                              ; 0741
                                      1D  12 001D7       BNEQ    29$                                 ;
         05             6B            01  E1 001D9       BBC     #1, EMPHASIS_BITS, 27$              ; 0745
                  0144  CB            53  D0 001DD       MOVL    K, BOLD_LIMIT                       ;
                                0138  CB  D5 001E2 27$:  TSTL    OVERSTRIKE_COUNT                    ; 0746
                                      05  13 001E6       BEQL    28$                                 ;
```

```
                 0148  CB              53 D0 001E8         MOVL    K, OVER_LIMIT                    ;
          05           6B              02 E1 001ED 28$:    BBC     #2, EMPHASIS_BITS, 29$           ; 0747
                 014C  CB              53 D0 001F1         MOVL    K, UNDER_LIMIT                   ;
                                       52 D4 001F6 29$:    CLRL    R2                               ; 0753
 04 00000000G  EF              04      04 ED 001F8         CMPZV   #4, #4, GCA+208, #4              ;
                                       02 12 00201         BNEQ    30$                              ;
                                       52 D6 00203         INCL    R2                               ;
                                       50 D4 00205 30$:    CLRL    R0                               ;
 05 00000000G  EF              04      04 ED 00207         CMPZV   #4, #4, GCA+208, #5              ;
                                       02 12 00210         BNEQ    31$                              ;
                                       50 D6 00212         INCL    R0                               ;
                               50      52 C8 00214 31$:    BISL2   R2, R0                           ;
                               01      50 D1 00217         CMPL    R0, #1                           ;
                                       21 12 0021A         BNEQ    32$                              ;
                                       51 DD 0021C         PUSHL   R1                               ; 0756
                               7E  013C CB 7D 0021E         MOVQ    OVERSTRIKE_CHAR, -(SP)          ;
                                   0138 CB DD 00223         PUSHL   OVERSTRIKE_COUNT                ; 0755
                                       5B DD 00227         PUSHL   R11                              ; 0754
 7E 00000000G  EF              01      00 EF 00229         EXTZV   #0, #1, GCA+209, -(SP)           ;
                                       55 DD 00232         PUSHL   HOLD_KHAR                        ;
             00000000G  EF             07 FB 00234         CALLS   #7, CNEMPH                       ;
                                       69 11 0023B         BRB     35$                              ;
                      01 00000000G  EF D1 0023D 32$:       CMPL    OUTOPT+12, #1                    ; 0768
                                       21 12 00244         BNEQ    33$                              ;
                                       51 DD 00246         PUSHL   R1                               ; 0771
                               7E  013C CB 7D 00248         MOVQ    OVERSTRIKE_CHAR, -(SP)          ;
                                   0138 CB DD 0024D         PUSHL   OVERSTRIKE_COUNT                ; 0770
                                       5B DD 00251         PUSHL   R11                              ; 0769
 7E 00000000G  EF              01      00 EF 00253         EXTZV   #0, #1, GCA+209, -(SP)           ;
                                       55 DD 0025C         PUSHL   HOLD_KHAR                        ;
             00000000G  EF             07 FB 0025E         CALLS   #7, BSEMPH                       ;
                                       3F 11 00265         BRB     35$                              ;
                      01 00000000G  EF D1 00267 33$:       CMPL    OUTOPT+16, #1                    ; 0773
                                       21 12 0026E         BNEQ    34$                              ;
                                       51 DD 00270         PUSHL   R1                               ; 0776
                               7E  013C CB 7D 00272         MOVQ    OVERSTRIKE_CHAR, -(SP)          ;
                                   0138 CB DD 00277         PUSHL   OVERSTRIKE_COUNT                ; 0775
                                       5B DD 0027B         PUSHL   R11                              ; 0774
 7E 00000000G  EF              01      00 EF 0027D         EXTZV   #0, #1, GCA+209, -(SP)           ;
                                       55 DD 00286         PUSHL   HOLD_KHAR                        ;
             00000000G  EF             07 FB 00288         CALLS   #7, OPEMPH                       ;
                                       15 11 0028F         BRB     35$                              ;
                                       0A DD 00291 34$:    PUSHL   #10                              ; 0779
                         00000000'  EF 9F 00293            PUSHAB  P.AAB                            ;
                         00000000G  8F DD 00299            PUSHL   #RNFILE                          ;
             00000000G  EF             03 FB 0029F         CALLS   #3, ERMS                         ;
                                       56 7C 002A6 35$:    CLRQ    OPERAND1                         ; 0783
                               6B      06 8A 002A8         BICB2   #6, EMPHASIS_BITS                ; 0786
                                   0138 CB 7C 002AB         CLRQ    OVERSTRIKE_COUNT                ; 0787
                                       61 11 002AF         BRB     42$                              ; 0737
                               01      51 D1 002B1 36$:    CMPL    R1, #1                           ; 0794
                                       05 15 002B4         BLEQ    37$                              ;
                               07      51 D1 002B6         CMPL    R1, #7                           ;
                                       44 19 002B9         BLSS    40$                              ;
 04 00000000G  EF              04      04 ED 002BB 37$:    CMPZV   #4, #4, GCA+208, #4              ; 0799
                                       0B 13 002C4         BEQL    38$                              ;
 05 00000000G  EF              04      04 ED 002C6         CMPZV   #4, #4, GCA+208, #5              ;
```

```
                                            25  12 002CF      BNEQ    39$                                ;
                            18              6B  93 002D1 38$: BITB    EMPHASIS_BITS, #24                 ; 0801
                                            20  13 002D4      BEQL    39$                                ;
                                            51  DD 002D6      PUSHL   R1                                 ; 0806
                            7E        013C  CB  7D 002D8      MOVQ    OVERSTRIKE_CHAR, -(SP)             ;
                                      0138  CB  DD 002DD      PUSHL   OVERSTRIKE_COUNT                   ; 0805
                                            5B  DD 002E1      PUSHL   R11                                ; 0804
       7E 00000000G  EF             01      00  EF 002E3      EXTZV   #0, #1, GCA+209, -(SP)             ;
                                    7E      01  CE 002EC      MNEGL   #1, -(SP)                          ;
                  00000000G  EF             07  FB 002EF      CALLS   #7, LNEMPH                         ;
                  00000000G  FF             55  90 002F6 39$: MOVB    HOLD_KHAR, @FRA+4                  ; 0825
                                            07  11 002FD      BRB     41$                                ;
                  00000000G  FF             20  90 002FF 40$: MOVB    #32, @FRA+4                        ; 0828
                                 00000000G  EF  D6 00306 41$: INCL    FRA+4                              ;
                                 00000000G  EF  D6 0030C      INCL    FRA+12                             ; 0825
     FDB6                53         01      59  F1 00312 42$: ACBL    R9, #1, K, 13$                     ; 0638
                            01        015C  CB  D1 00318      CMPL    PASS_CNTR, #1                      ; 0837
                                            1F  12 0031D      BNEQ    46$                                ;
                                      0148  CB  D5 0031F      TSTL    OVER_LIMIT                         ; 0840
                                            0C  12 00323      BNEQ    43$                                ;
                                      014C  CB  D5 00325      TSTL    UNDER_LIMIT                        ; 0841
                                            06  12 00329      BNEQ    43$                                ;
                                      0144  CB  D5 0032B      TSTL    BOLD_LIMIT                         ; 0842
                                            05  13 0032F      BEQL    44$                                ;
                            50              07  D0 00331 43$: MOVL    #7, R0                             ; 0840
                                            03  11 00334      BRB     45$                                ;
                            50              01  D0 00336 44$: MOVL    #1, R0                             ;
                       0158 CB              50  D0 00339 45$: MOVL    R0, LAST_PASS                      ;
       04 00000000G  EF     04              04  ED 0033E 46$: CMPZV   #4, #4, GCA+208, #4                ; 0849
                                            0B  13 00347      BEQL    47$                                ;
       05 00000000G  EF     04              04  ED 00349      CMPZV   #4, #4, GCA+208, #5                ;
                                            27  12 00352      BNEQ    48$                                ;
                            18              6B  93 00354 47$: BITB    EMPHASIS_BITS, #24                 ;
                                            22  13 00357      BEQL    48$                                ;
                                      015C  CB  DD 00359      PUSHL   PASS_CNTR                          ; 0854
                            7E        013C  CB  7D 0035D      MOVQ    OVERSTRIKE_CHAR, -(SP)             ;
                                      0138  CB  DD 00362      PUSHL   OVERSTRIKE_COUNT                   ; 0853
                                            5B  DD 00366      PUSHL   R11                                ; 0852
       7E 00000000G  EF             01      00  EF 00368      EXTZV   #0, #1, GCA+209, -(SP)             ;
                                    7E      01  CE 00371      MNEGL   #1, -(SP)                          ;
                  00000000G  EF             07  FB 00374      CALLS   #7, LNEMPH                         ;
                            09 00000000G    EF  E8 0037B 48$: BLBS    OUTOPT+12, 49$                     ; 0890
                       0158 CB        015C  CB  D1 00382      CMPL    PASS_CNTR, LAST_PASS               ; 0892
                                            04  12 00389      BNEQ    50$                                ;
                            50              01  D0 0038B 49$: MOVL    #1, R0                             ; 0894
                                                04 0038E      RET                                        ;
                            50        015C  CB  D0 0038F 50$: MOVL    PASS_CNTR, R0                      ; 0898
                            01              50  D1 00394      CMPL    R0, #1                             ;
                                            18  15 00397      BLEQ    51$                                ;
                            07              50  D1 00399      CMPL    R0, #7                             ;
                                            13  18 0039C      BGEQ    51$                                ;
                  00000000G  FF             0D  90 0039E      MOVB    #13, @FRA+4                        ; 0900
                                 00000000G  EF  D6 003A5      INCL    FRA+4                              ;
                                 00000000G  EF  D6 003AB      INCL    FRA+12                             ;
                            63              50  E8 003B1 51$: BLBS    R0, 59$                            ; 0915
                            51        0158  CB  D0 003B4      MOVL    LAST_PASS, R1                      ; 0916
                                            1F  15 003B9      BLEQ    55$                                ;
```

```
                                          52   D4 003BB          CLRL    R2                              ;
                             03           50   D1 003BD          CMPL    R0, #3                          ;
                                          04   12 003C0          BNEQ    52$                             ;
                                          52   D6 003C2          INCL    R2                              ;
                                          05   11 003C4          BRB     53$                             ;
                             04           50   D1 003C6 52$:     CMPL    R0, #4                          ;
                                          1E   12 003C9          BNEQ    56$                             ;
                             05           52   E8 003CB 53$:     BLBS    R2, 54$                         ;
                             04           50   D1 003CE          CMPL    R0, #4                          ;
                                          07   12 003D1          BNEQ    55$                             ;
                             0F 00000000G EF   E9 003D3 54$:     BLBC    OUTOPT+8, 56$                   ;
                                          51   D5 003DA 55$:     TSTL    R1                              ;
                                          39   18 003DC          BGEQ    59$                             ;
                             50 00000000G EF   D0 003DE          MOVL    TSF, R0                         ;
                             2E        08 A0   E9 003E5          BLBC    8(R0), 59$                      ;
          04 00000000G  EF   04           04   ED 003E9 56$:     CMPZV   #4, #4, GCA+208, #4             ; 0919
                                          23   13 003F2          BEQL    59$                             ;
          05 00000000G  EF   04           04   ED 003F4          CMPZV   #4, #4, GCA+208, #5             ;
                                          18   13 003FD          BEQL    59$                             ;
                             53 00000000G EF   D0 003FF          MOVL    OUTOPT+20, R3                   ; 0921
                                          52   D4 00406          CLRL    I                               ;
                                          09   11 00408          BRB     58$                             ;
                                          0B   DD 0040A 57$:     PUSHL   #11                             ; 0922
                      00000000G  EF       01   FB 0040C          CALLS   #1, CLH                         ;
                   F3            52       53   F2 00413 58$:     AOBLSS  R3, I, 57$                      ;
                                 50  015C CB   D0 00417 59$:     MOVL    PASS_CNTR, R0                   ; 0926
                             01           50   D1 0041C          CMPL    R0, #1                          ;
                                          59   15 0041F          BLEQ    65$                             ;
                             07           50   D1 00421          CMPL    R0, #7                          ;
                                          54   18 00424          BGEQ    65$                             ;
                             51      0158 CB   D0 00426          MOVL    LAST_PASS, R1                   ; 0927
                                          1F   15 0042B          BLEQ    63$                             ;
                                          52   D4 0042D          CLRL    R2                              ;
                             03           50   D1 0042F          CMPL    R0, #3                          ;
                                          04   12 00432          BNEQ    60$                             ;
                                          52   D6 00434          INCL    R2                              ;
                                          05   11 00436          BRB     61$                             ;
                             04           50   D1 00438 60$:     CMPL    R0, #4                          ;
                                          1E   12 0043B          BNEQ    64$                             ;
                             05           52   E8 0043D 61$:     BLBS    R2, 62$                         ;
                             04           50   D1 00440          CMPL    R0, #4                          ;
                                          07   12 00443          BNEQ    63$                             ;
                             0F 00000000G EF   E9 00445 62$:     BLBC    OUTOPT+8, 64$                   ;
                                          51   D5 0044C 63$:     TSTL    R1                              ;
                                          2A   18 0044E          BGEQ    65$                             ;
                             50 00000000G EF   D0 00450          MOVL    TSF, R0                         ;
                             1F        08 A0   E9 00457          BLBC    8(R0), 65$                      ;
          04 00000000G  EF   04           04   ED 0045B 64$:     CMPZV   #4, #4, GCA+208, #4             ; 0930
                                          14   13 00464          BEQL    65$                             ;
          05 00000000G  EF   04           04   ED 00466          CMPZV   #4, #4, GCA+208, #5             ;
                                          09   13 0046F          BEQL    65$                             ;
                                          0B   DD 00471          PUSHL   #11                             ; 0932
                      00000000G  EF       01   FB 00473          CALLS   #1, CLH                         ;
          04 00000000G  EF   04           04   ED 0047A 65$:     CMPZV   #4, #4, GCA+208, #4             ; 0940
                                          0B   13 00483          BEQL    66$                             ;
          05 00000000G  EF   04           04   ED 00485          CMPZV   #4, #4, GCA+208, #5             ;
                                          09   12 0048E          BNEQ    67$                             ;
```

```
                                05      015C    CB  D1 00490 66$:    CMPL    PASS_CNTR, #5                           ; 0945
                                                24  18 00495        BGEQ    69$                                     ;
                                                06  11 00497        BRB     68$                                     ; 0947
                                        0158    CB  D5 00499 67$:    TSTL    LAST_PASS                               ; 0951
                                                1C  15 0049D        BLEQ    69$                                     ;
                                    00000000G   EF  D4 0049F 68$:    CLRL    FRA+12                                  ; 0953
                        00000000G   EF  00000000G   EF  9E 004A5        MOVAB   FRA+16, FRA                             ;
                        00000000G   EF  00000000G   EF  D0 004B0        MOVL    FRA, FRA+4                              ;
                        00000000V   EF              00  FB 004BB 69$:    CALLS   #0, COMPUTE_NEXT_PASS                   ; 0958
                            015C    CB              50  D0 004C2        MOVL    R0, PASS_CNTR                           ;
                                                    50  D4 004C7        CLRL    R0                                      ; 0962
                                                        04 004C9        RET                                             ;

; Routine Size:  1226 bytes,    Routine Base:  $CODE$ + 0182
```

```
1 %SBTTL 'compute_next_pass -- Decide which pass comes next'
1 ROUTINE compute_next_pass =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Decide which pass over the MRA the next pass should be.
1 !
1 ! FORMAL PARAMETERS:    None
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       The value of the OWN variable pass_cntr is used as the starting
1 !       point for the calculation.
1 !
1 !       Values in the tsf, outopt, and gca structures are used (in some
1 !       cases by way of macros defined at the top of this module).
1 !
1 !       The order of the pass_... literals defined in PASS.REQ determines
1 !       the overall logic of this routine.
1 !
1 ! IMPLICIT OUTPUTS:     None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Returns the new value for pass_cntr (but does not update the OWN
1 !       itself).
1 !
1 ! SIDE EFFECTS:         None
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         next_one;
2
2     ! Start with the current value of the pass counter.
2     next_one = .pass_cntr;
2
2     ! Increment based on whether there is any bolding or not.
3     next_one = (IF .tsf_bld
3                 THEN
4                     (.next_one + 1)
3                 ELSE
4                     (.next_one + 2)
2                 );
2
2     ! Check for underlining in Passes 3 and 4.
3     IF  (.next_one EQL pass_underline)
2         OR
3         (.next_one EQL pass_bold_underline)
2     THEN
3         IF (NOT .tsf_und)                   !No underlining to do, or
3           OR (.outopt_und_nosp              ! non-spacing underline
3               AND NOT .outopt_und_sep)      ! (was already done)...
```

```
2           THEN
2               next_one = .next_one + 2;    ! Skip the underlining pass.
2
2       ! For LN01 output, any overstriking is taken care of at the
2       ! overstrike pass (only).
3       IF  (.next_one EQL pass_bold_overstrike)
2           AND
3           (laser_output)
2       THEN
2           next_one = pass_real_text;
2
2       ! Skip if no overstriking required.
3       IF  (.next_one GEQ pass_overstrike)
2           AND
3           (NOT .tsf_ovr)
2       THEN
2           next_one = pass_real_text;
2
2       RETURN .next_one;
2
1       END;                                    ! End of compute_next_pass
```

```
                                      0004 00000 COMPUTE_NEXT_PASS:
                                                       .WORD   Save R2                        ; 0964
                   52 00000000G  EF  9E 00002          MOVAB   GCA+208, R2                    ;
                   51 00000000'  EF  D0 00009          MOVL    PASS_CNTR, NEXT_ONE            ; 1002
                   50 00000000G  EF  D0 00010          MOVL    TSF, R0                        ; 1005
                   04        08  A0  E9 00017          BLBC    8(R0), 1$                      ;
                                 51  D6 0001B          INCL    NEXT_ONE                       ; 1007
                                 03  11 0001D          BRB     2$                             ;
                   51            02  C0 0001F 1$:      ADDL2   #2, NEXT_ONE                   ; 1009
                   03            51  D1 00022 2$:      CMPL    NEXT_ONE, #3                   ; 1013
                                 05  13 00025          BEQL    3$                             ;
                   04            51  D1 00027          CMPL    NEXT_ONE, #4                   ; 1015
                                 16  12 0002A          BNEQ    5$                             ;
         0E        08  A0        01  E1 0002C 3$:      BBC     #1, 8(R0), 4$                  ; 1017
                   0A 00000000G  EF  E9 00031          BLBC    OUTOPT+4, 5$                   ; 1018
                   03 00000000G  EF  E8 00038          BLBS    OUTOPT+8, 5$                   ; 1019
                   51            02  C0 0003F 4$:      ADDL2   #2, NEXT_ONE                   ; 1021
                   06            51  D1 00042 5$:      CMPL    NEXT_ONE, #6                   ; 1025
                                 11  12 00045          BNEQ    7$                             ;
   04      62      04            04  ED 00047          CMPZV   #4, #4, GCA+208, #4            ; 1027
                                 07  13 0004C          BEQL    6$                             ;
   05      62      04            04  ED 0004E          CMPZV   #4, #4, GCA+208, #5            ;
                                 03  12 00053          BNEQ    7$                             ;
                   51            07  D0 00055 6$:      MOVL    #7, NEXT_ONE                   ; 1029
                   05            51  D1 00058 7$:      CMPL    NEXT_ONE, #5                   ; 1032
                                 08  19 0005B          BLSS    8$                             ;
         03        08  A0        02  E0 0005D          BBS     #2, 8(R0), 8$                  ; 1034
                   51            07  D0 00062          MOVL    #7, NEXT_ONE                   ; 1036
                   50            51  D0 00065 8$:      MOVL    NEXT_ONE, R0                   ; 1038
                                     04 00068          RET                                    ; 1040
```

```
; Routine Size:  105 bytes,    Routine Base:  $CODE$ + 064C


;  916          1041  1
;  917          1042  1 END                                                           ! End of module
;  918          1043  0 ELUDOM




;                               PSECT SUMMARY
;
;       Name                          Bytes                        Attributes
;
;     $OWN$                             352  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;     $CODE$                           1717  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;     $PLIT$                             20  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                          Library Statistics
;
;                                                  -------- Symbols --------        Pages         Processing
;       File                                       Total    Loaded   Percent        Mapped        Time
;
;     _$255$DUA28:[SYSLIB]XPORT.L32;1                590         0         0          252          00:00.1
;     _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1          1248        67         5           86          00:00.2




;                               COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LOHORI/OBJ=OBJ$:LOHORI MSRC$:LOHORI/UPDATE=(ENH$:LOHORI)

; Size:          1717 code + 372 data bytes
; Run Time:         00:27.2
; Elapsed Time:     01:00.6
; Lines/CPU Min:     2304
; Lexemes/CPU-Min: 17293
; Memory Used:  259 pages
; Compilation Complete
```